AMENDMENTS TO THE CLAIMS

D1

1. (Previously Amended) A method to process commands in a computer memory subsystem, comprising:
   (a) receiving a plurality of commands on a bus network connected to said memory subsystem;
   (b) categorizing said received commands into command types;
   (c) placing each received command into a queue pertaining to its respective command type;
   (d) determining memory cycle performance penalties of said categorized commands in each of said queues;
   (e) reordering said categorized commands in each of said queues so that one categorized command in each of said queues having the least memory cycle performance penalty is selected for execution;
   (f) determining if each of said selected command is valid;
   (g) arbitrating said valid commands and selecting one of said valid commands to execute; and
   (h) executing sequential valid commands of the same command type.

2. (Original) The method of claim 1, wherein said command types are forms of store and fetch operations.

3. (Original) The method of claim 1, wherein said command types are associated with a particular source or destination of said received memory commands.

D1

4. (Original) The method of claim 3, wherein said particular source or destination is a particular computer processor connected on said bus network.

5. (Original) The method of claim 3, wherein said particular source or destination is a I/0 hub controller functionally connected on said bus network.

6. (Original) The method of claim 3, wherein said particular source or destination is a switching fabric connected to said bus network.

7. (Original) The method of claim 3, wherein said particular source or destination is a compression/decompression engine functionally connected to said bus network.

8. (Original) The method of claim 1, wherein said command types which originate from or are required for a particular application have priority.

9. (Original) The method of claim 1, wherein said step of receiving a plurality of commands further comprises determining if any of said received commands have an address dependency and passing said address dependency determination with said memory command.

D1

10. (Original) The method of claim 1, wherein said step of determining memory cycle performance penalties of said categorized commands further comprises comparing a number of oldest received categorized commands with each other.

11. (Original) The method of claim 9, wherein said step of determining memory cycle performance penalties of said categorized commands further comprises comparing a number of the oldest received categorized commands with a currently chosen command.

12. (Original) The method of claim 9, wherein said step of determining memory cycle performance penalties of said categorized commands further comprises comparing a number of the oldest received categorized commands with a previously chosen command.

13. (Original) The method of claim 1, wherein said step of reordering said categorized commands further comprises selecting the oldest of said categorized commands that have the least memory cycle performance penalty for execution.

D1

14. (Original) The method of claim 1, wherein said step of arbitrating said reordered valid commands further comprises granting priority to said type of command having said least memory cycle performance penalty.

15. (Original) The method of claim 1, wherein said step of arbitrating said reordered valid commands further comprises granting priority to a command type other than said command type of said reordered valid commands.

16. (Previously Amended) The method of claim 1, wherein said step of executing sequential valid commands of the same command type further continues until a valid memory command of said command type is no longer available, or until a predetermined number has been executed, or until a memory command of another of said command types has higher priority.

D1

17. (Previously Amended) A method to process commands in a computer memory subsystem, comprising:

(a) receiving a plurality of memory commands on a bus connected to said computer memory subsystem and determining the physical location of the memory command in memory, and further determining if any of said received memory commands have an address dependency and passing said physical location and said address dependency, if any, corresponding to said memory command along with said memory command;

(b) categorizing said received commands into command types based on one of the following: STORE, FETCH, INTERVENTION STORE; the source or destination of said received memory commands; the program or application from which said memory commands originate or are otherwise required;

(c) determining memory cycle performance penalties of said categorized commands by comparing a number of oldest received categorized commands with each other, with a currently chosen command, and with a previously chosen command;

(d) reordering said categorized commands so that said categorized commands having the least memory cycle performance penalty are selected for execution and if more than one categorized command has the least memory cycle performance penalty, then selecting the oldest of said reordered commands for execution;

(e) determining if said reordered commands are valid;

D1

(f) granting priority to said type of command having said least memory cycle performance penalty;

(g) executing sequential valid commands of the same command type until a valid command of the same type is not received or until a predetermined number has been executed, or until a memory command of another type has higher priority;

(h) avoiding deadlock when an address dependency exists between commands of different types by executing commands having the command type of the oldest memory command.

18. (Original) A method of processing memory commands in a computer processing system having at least one command source on a bus connected to a memory controller, said method comprising selecting a memory command having the least memory cycle performance penalties to execute and then executing a programmable number of other memory commands of that type.

19. (Currently Amended) A computer processing system, comprising:

(a) a plurality of bus units, said bus units comprising at least one computer processor, at least one I/O device; at least one memory cache system connected to said at least one computer processor, and at least one network communication device, said plurality of bus units interconnected on a bus network, and said plurality of bus units to issue memory commands, said memory commands categorized into types;

(b) at least one memory subsystem connected on a first bus to said plurality of bus units, said memory subsystem responsive to said memory commands and further comprising:

(i) a memory controller connected to a command interface functionally connected to said first bus;

(ii) a plurality of memory chips configured into memory banks; said memory chips architected into memory cards attached to at least one memory bus;

(iii) a plurality of command FIFO queues, each of said command FIFO queues associated with one of said command types into which said memory commands are categorized;

(iv) a plurality of comparison logic circuits, each of said plurality of comparison logic circuits associated with each of said plurality of command FIFO queues to determine which memory commands of each of said command types have the least memory cycle performance penalty by comparing a number of oldest received categorized commands with each other, with a currently chosen command, and with a previously chosen command;

D1

(v) an arbitration logic circuit to output said memory commands of said determined command type having said least memory cycle performance penalty to said plurality of memory chips.

20. (Original) The computer processing system of claim 19, wherein said comparison logic circuit further determines the oldest of said memory commands in each of said plurality of command FIFO queues.

21. (Currently Amended) A computer memory controller, comprising:

(a) means to receive a plurality of types of memory commands from a plurality of command sources;

(b) means to determine the memory cycle performance penalty associated with each memory command of each of said plurality of types;

(c) means to compare said memory commands of ~~one~~ each of said types with other memory commands of the same type to determine which of said memory commands have the least memory cycle performance penalty;

D1 end

(d) means to compare said memory commands of ~~one~~ each of said types with a current chosen memory command of the same type to determine which of said memory commands have the least memory cycle performance penalty;

(e) means to compare said memory commands of ~~one~~ each of said types with a previously chosen memory command of the same type determine which of said memory commands have the memory cycle performance penalty;

(f) means to select one of said memory commands having the least memory cycle performance penalty by selecting the oldest; and

(g) means to continue execution of memory commands of the same type as said selected memory command.